**Documentación adicional: (SOLO ESTUDIANTES DE CE)**

**1. Atributo de Análisis de Problema**

**(a) Identificación del problema complejo de ingeniería**

El problema complejo de ingeniería planteado consiste en diseñar e implementar un videojuego funcional, interactivo y escalable que replique las mecánicas esenciales del clásico *Bomberman*. Este reto requiere aplicar principios de **matemáticas discretas**, **lógica computacional**, **física básica** (para manejo de colisiones y efectos temporales), y fundamentos de **programación orientada a objetos (POO)**. Además, se relaciona con el desarrollo sostenible al promover la creación de soluciones tecnológicas educativas y recreativas de bajo impacto ambiental, usando tecnologías de código abierto como **Python**.

**(b) Análisis del contexto y variables relacionadas**

En el contexto actual, la industria del software educativo y de entretenimiento representa una oportunidad significativa para desarrollar habilidades de programación. Las variables clave incluyen:

* **Rendimiento gráfico en tiempo real** (responsivo en GUI).
* **Modularidad del código** (POO).
* **Accesibilidad del juego** (uso de bibliotecas gratuitas como Pygame).
* **Jugabilidad escalonada** (niveles con aumento de dificultad).
* **Educación sostenible**: el videojuego actúa como medio formativo que promueve el aprendizaje activo.

Estos factores deben armonizarse sin comprometer los principios del desarrollo sostenible: reutilización de código, eficiencia en el uso de recursos computacionales, y fomento del aprendizaje en entornos accesibles.

**(c) Plan de solución**

El plan de solución incluye:

1. **Análisis del problema**: descomponer el juego en clases (jugador, enemigo, bomba, mapa, GUI).
2. **Diseño modular**: definir estructuras de datos reutilizables y escalables.
3. **Desarrollo iterativo**: emplear ciclos de desarrollo cortos para implementar y probar funciones por separado.
4. **Interfaz gráfica**: desarrollo con Pygame para garantizar interacción visual y eventos en tiempo real.
5. **Gestión sostenible del proyecto**: planificación de tareas, uso de herramientas de control de versiones y documentación clara.

**(d) Evaluación de las soluciones**

| **Solución propuesta** | **Pros** | **Contras** |
| --- | --- | --- |
| Implementación en Pygame | Permite mayor fluidez y personalización visual | Mayor curva de aprendizaje, más consumo de recursos |
| Programación Orientada a Objetos (POO) | Favorece la escalabilidad y mantenimiento del código | Requiere una buena estructura inicial para evitar sobreacoplamientos |
| Juego por niveles escalables | Aumenta el interés del jugador, permite ajustes de dificultad | Mayor esfuerzo en balancear niveles y pruebas |
| Incorporación de elementos aleatorios | Rejugabilidad, dinámica variada | Requiere mayor control de condiciones límite y errores lógicos posibles |

**2. Atributo de Herramientas de Ingeniería**

**(a) Técnicas, recursos, herramientas o métodos utilizados**

Para resolver el problema se utilizan las siguientes herramientas y técnicas:

* **Python 3.11**: lenguaje principal, orientado a objetos.
* **Pygame**: para la interfaz gráfica.
* **POO**: organización modular mediante clases.
* **Control de versiones con Git**: para el trabajo colaborativo.
* **Algoritmos de colisión y eventos temporales**: manejo de explosiones, movimientos y trampas.
* **Manejo de estructuras de datos**: listas, diccionarios y matrices para representar el mapa.

**(b) Aplicación de las herramientas en el proyecto**

Cada recurso se aplica de la siguiente forma:

* **Pygame**: renderización de personajes, interfaz de usuario, animaciones, y eventos de teclado.
* **POO**: implementación de clases Jugador, Enemigo, Bomba, Mapa, etc., cada una con atributos y métodos.
* **Algoritmos de búsqueda** (ej. BFS para propagación de explosión): simulan interacciones entre bombas y obstáculos.
* **GitHub**: control de versiones, sincronización de cambios, documentación con archivos README.

**(c) Adaptación de las herramientas en el desarrollo**

Se han adaptado las herramientas de la siguiente manera:

* **Pygame** fue personalizado para trabajar con *sprites* estáticos a través de transformaciones y ImageLoad, lo que permite una interfaz ligera y flexible.
* **Git** se adaptó al flujo del curso dividiendo las tareas por clases y commits semánticos por funcionalidad.
* **Estructuras de datos** fueron adaptadas para representar el mapa como una lista de listas (matriz), facilitando el recorrido y el rastreo de objetos.

**3. Literatura o Fuentes Consultadas**

1. Lutz, M. (2013). *Learning Python* (5th ed.). O’Reilly Media.
2. Sweigart, A. (2015). *Making Games with Python & Pygame*. InventWithPython.com.
3. Tkinter documentation - <https://docs.python.org/3/library/tkinter.html>
4. Pygame documentation - https://www.pygame.org/docs/
5. MIT OpenCourseWare – Introduction to Computer Science and Programming in Python.
6. Instituto Tecnológico de Costa Rica – Documento oficial del proyecto: *Proyecto I – Vintage Bomberman Game* (I Semestre 2025).
7. Guía UNESCO: *Desarrollo sostenible a través de la tecnología y la educación* (2019).